A Privacy-Preserving Architecture for Secure Appliance-Level Demand Response in Virtual Power Plants

Joan Ferré-Queralt[1][0009−0002−0014−0065], Jordi Castellà-Roca[1][0000−0002−0037−9888], Alexandre Viejo[1][0000−0003−2342−5100], Nicolás Orduz Álvarez[2][0009−0006−3776−7987], and Takao Tsuji[2]

[1] Universitat Rovira i Virgili
Departament d'Enginyeria Informàtica i Matemàtiques
CYBERCAT-Center for Cybersecurity Research of Catalonia
Av. Països Catalans 26, E-43007 Tarragona, Catalonia, Spain
{joan.ferreq,jordi.castella,alexandre.viejo}@urv.cat
[2] Yokohama National University
Graduate School of Engineering Science
Yokohama, Japan
nicolas-alvarez-zn@ynu.jp,tsuji-takao-pz@ynu.ac.jp

Abstract. Virtual Power Plants (VPPs) enable the aggregation of small flexible loads, allowing residential appliances to contribute to demand response (DR) programs. However, this participation requires fine-grained information about appliance availability, operational constraints, and user preferences, and may also involve remote appliance activation. As a result, appliance-level DR introduces privacy and security risks that are not sufficiently addressed by conventional aggregation approaches. In particular, availability reports, activation decisions, and delivery evidence may reveal sensitive behavioral patterns or enable unauthorized manipulation of DR operations. This paper proposes a privacy-preserving and secure architecture for appliance-level DR in VPPs. The framework combines appliance clustering, OpenADR-based activation, public-key cryptography, operational pseudonyms, and smart contracts to support verifiable DR coordination while reducing the exposure of user-specific information. The proposed design protects appliance availability from public disclosure, prevents unauthorized observers from linking DR actions to real users, and enables authenticated activation and evidence verification. A qualitative evaluation shows that the architecture satisfies the defined functional, privacy, and security requirements, providing a practical basis for secure and privacy-aware appliance participation in VPP-based DR programs.

Keywords: Demand response · Virtual Power Plant · Privacy · Security · Smart contracts

1 Introduction

Electric power systems must continuously maintain the balance between generation and demand. Imbalances may compromise system stability and lead to blackouts [7]. Traditionally, demand has had peaks and valleys that do not always coincide with the forecasting and this challenge is now becoming more acute. Electricity consumption continues to increase, placing additional stress on grid operation, while renewable energy resources have reached unprecedented levels of penetration [21]. These zero-emission sources are less predictable, which introduces less controllable fluctuations in power supply [11].

As a result, grid stability increasingly depends on robust ancillary services and advanced demand response (DR) mechanisms capable of preserving reliability.

Historically, DR programs focused mainly on large industrial and commercial consumers with significant load-reduction capacity. Early schemes were commonly based on interruptibility contracts or incentive-based mechanisms, in which participants agreed to reduce consumption when requested by the system operator [23]. This model offered system operators relatively large and reliable reductions from a limited number of participants. However, such programs provide limited granularity. Since flexibility is obtained from a small number of large consumers, the operator can only control demand at an aggregated and relatively coarse level.

To overcome these limitations, research has increasingly considered the use of smaller distributed loads as providers of ancillary services [6]. Some examples are residential appliances and other user-side resources, which have limited individual capacity, but their aggregate flexibility can become substantial when coordinated at scale.

Individual appliances generally cannot participate directly in service markets, where minimum capacity thresholds are required. In response, Virtual Power Plants (VPPs) have emerged as aggregators that combine residential and commercial resources into a larger virtual resource capable of satisfying market participation requirements [13]. VPPs must solve scheduling problems over heterogeneous loads by classifying them into similar (in terms of power and constraints) clusters and treating each cluster as one load.

Moreover, DR participation requires the exchange of fine-grained flexibility information and may involve remote control over user appliances. Flexibility data can reveal sensitive behavioral patterns of users, such as occupancy, daily routines, appliance usage and user preferences. At the same time, remote activation introduces security risks, since forged commands, manipulated schedules, or fraudulent participation evidence may affect user comfort, remuneration, and even grid operation. Consequently, security and privacy must be considered fundamental design requirements in any VPP-based demand response framework.

1.1 Related Work

Authors in [10, 14, 16, 18] proposed optimal demand aggregation frameworks in which VPPs schedule clustered flexible loads for market participation. However, these approaches provide only limited treatment of user-level comfort constraints. Other works [1, 4, 5, 17, 20] address such constraints while introducing different VPP load scheduling algorithms.

Nevertheless, these works generally assume that the VPP already has access to reliable information about the flexibility, availability, and operational constraints of participating loads. Before any DR activation can be scheduled, this information must first be collected. Since appliance-level availability and constraints may reveal sensitive information about users' habits, privacy must be considered from the design stage.

Privacy in DR systems involving user appliances as deferrable loads has received limited attention. Some works rely on cryptographic techniques, such as homomorphic encryption or Shamir's Secret Sharing, to allow appliance-related data to be shared with aggregators while limiting the disclosure of individual user information [8,15,22]. Other approaches [3,9,19] use consortium blockchain-based frameworks to preserve user privacy and maintain an immutable record of system operations.

However, these contributions typically address isolated parts of the DR workflow, such as privacy-preserving data aggregation or auditable transaction recording, rather than the complete appliance-level DR process. In particular, they do not jointly consider appliance setup and registration, the collection of fine-grained availability and operational constraints, the aggregation of appliances into schedulable flexibility clusters, the secure activation of end-user loads by a VPP, and the generation of verifiable evidence for remuneration. Moreover, when appliances are directly controllable by a third party, additional security issues arise, including activation authenticity, unauthorized control, evidence forgery, and the linkability of activation or delivery records to real users. These aspects remain insufficiently addressed in existing privacy-preserving DR and blockchain-based VPP proposals.

1.2 Contributions

This paper addresses the gap in the literature by proposing a privacy-preserving DR architecture for interoperable appliance-level aggregation, scheduling, activation, verification, and remuneration under VPP coordination. The main contributions of this work are:

- Propose a DR architecture for VPP-based appliance-level flexibility management, where residential and commercial appliances are registered, clustered, scheduled, activated, and remunerated through a unified framework.
- Introduce a privacy-preserving availability reporting mechanism, allowing the VPP to compute cluster-level flexibility without public disclosure of individual availability patterns.

- Define a secure activation in which appliances accept only authenticated events from the VPP.
- Design an evidence-verification procedure in which compensation is released only for signed evidence while preserving the privacy of activated appliances.

2 System Overview

The overall architecture of the proposed system is presented in this section. It introduces the main actors and components, summarizes their responsibilities, and defines the appliance-level parameters used throughout the protocol.

2.1 Actors and Components

The proposed framework involves several actors and technical components, each playing a specific role in the aggregation, scheduling, activation, and verification of demand response (DR) services.

Users A user U_i denotes an individual or entity enrolled in a DR program through a contractual agreement with a VPP. Each user owns a set of controllable appliances that can provide flexible power curtailment by adapting their operation in response to external DR signals. Through a mobile application, the user registers appliances, configures their operational preferences, and defines availability and comfort-related constraints.

Appliances Each appliance $a_{i,j}$ is owned by user U_i and represents a controllable load that can participate in the DR program. Appliances are characterized by a set of technical and operational parameters that define their curtailment capacity $P_{i,j}$ and user-defined constraints $\theta_{i,j} = (max_{i,j},\ rec_{i,j})$.

Where $P_{i,j}$ denotes the nominal power of appliance $a_{i,j}$, $max_{i,j}$ denotes the maximum time that appliance $a_{i,j}$ must remain in OFF state, and $rec_{i,j}$ denotes the recovery time required after a curtailment event before appliance $a_{i,j}$ can participate in another DR activation.

We assume that each participating appliance contains a trusted control and measurement component connected to a hardware secure module (HSM). The HSM releases an operational signature for delivery evidence only if the appliance control logic verifies that the requested curtailment state was maintained during the activation interval.

Manufacturers Each manufacturer M_m is responsible for producing and provisioning appliances that can participate in the proposed DR framework. Manufacturers are trusted to equip appliances with secure hardware components. Manufacturers also provide certified information about appliance capabilities, including the nominal power value used during registration and clustering.

Clusters To perform the scheduling [12], the VPP groups appliances with similar curtailment power and operational constraints. Each cluster c is represented by $\theta_c = (P_c, \max_c, \mathrm{rec}_c)$, where P_c is a guaranteed per-appliance curtailable power, rather than the total cluster capacity. The parameters are conservatively selected so that every appliance assigned to the cluster satisfies

$$P_{i,j} \geq P_c, \qquad \max_{i,j} \geq \max_c, \qquad \mathrm{rec}_{i,j} \leq \mathrm{rec}_c.$$

Equivalently, when computed from a fixed set of members, these values can be selected as the minimum curtailable power, minimum maximum-OFF time, and maximum recovery time, respectively. Hence, $N_{c,d}[t]P_c$ is a conservative estimate of the available cluster capacity.

The VPP defines the admissible ranges of these parameters for each cluster in a step previous to the setup. During registration, the RA compares $(P_{i,j}, \theta_{i,j})$ with these predefined cluster profiles and assigns $a_{i,j}$ to a compatible cluster through $C_c \leftarrow \mathrm{AssignCluster}(P_{i,j}, \theta_{i,j})$. The particular partitioning of the parameter space is deployment-specific; the proposed security and privacy mechanisms are independent of the clustering algorithm used.

Cluster Smart Contract ($\mathcal{CSC}_c$) For each cluster c, the VPP deploys a cluster smart contract $\mathcal{CSC}_c$, which stores the cluster parameters and supports the later availability reporting, scheduling, activation, evidence verification, and payment procedures.

The contract therefore mainly performs authorization, state lookup, signature verification, replay checks, and payment accounting. Using key-value mappings, these checks have constant storage-access complexity per transaction. However, the total on-chain storage and transaction volume grow with the number of participating appliances and scheduled addresses. In large deployments, the scheduled set $\alpha_{c,d,I}$ could instead be represented by a Merkle-root commitment, with appliances providing membership proofs during evidence submission, reducing the on-chain schedule representation from $O(|\alpha_{c,d,I}|)$ to $O(1)$.

Virtual Power Plant The Virtual Power Plant (VPP) aggregates flexibility from clusters. This enables the VPP to treat a set of appliances as if it were only one load. The VPP also collects availability information, performs scheduling and issues DR activation commands. In the proposed architecture, the VPP acts as the main coordinator of the DR program and interfaces with electricity markets as an aggregated flexibility provider.

Registration Authority The Registration Authority (RA) verifies appliance registration information, assigns appliances to the appropriate flexibility clusters, and authorizes them to report curtailment availability.

2.2 Phases

The protocol is organized into three main phases: the *setup phase*, the *registration phase*, and the recurring *operational phase*.

Setup phase. During setup, the VPP, manufacturers, and system authorities generate and distribute the cryptographic material required by the protocol. The VPP defines the set of flexibility clusters and deploys the corresponding cluster smart contracts $\mathcal{CSC}_c$.

Registration phase. During appliance registration, each user U_i authenticates with the Registration Authority (RA) and enrolls their appliances $a_{i,j}$ in the DR program. For each appliance, the user provides the required operational preferences and constraints. Based on the appliance profile, the RA assigns the appliance to a flexibility cluster c.

Operational phase. After registration, the operational phase is executed periodically and consists of the following steps, which are also shown in Figure 1:

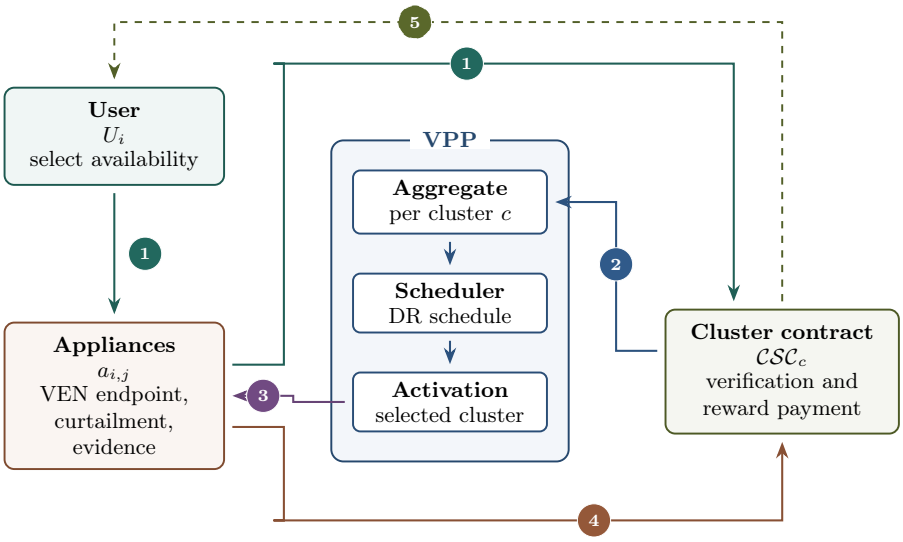


Fig. 1. Architecture-level view of the recurring scheduling phase without a gateway.

– *Step 1 - Availability submission:* Each user U_i specifies the daily curtailment availability of their appliances $a_{i,j}$.

- *Step 2 - Scheduling:* The VPP aggregates the availability reported for each cluster c and computes the corresponding DR schedule.
- *Step 3 - Activation:* When a DR event must be executed, the VPP activates the corresponding cluster. The activation signal is sent to the appliances $a_{i,j}$ that belong to the activated cluster.
- *Step 4 - Evidence generation:* Each scheduled and activated appliance $a_{i,j}$ performs the requested curtailment according to its operational constraints and generates evidence of activation. This evidence is submitted for verification.
- *Step 5 - Verification and payment:* The submitted activation evidence is verified. If the evidence is valid and the claimed response satisfies the activation requirements, the contract authorizes the corresponding remuneration.

2.3 Adversary Model and Requirements

The adversary model and requirements are presented in this section.

Adversary Model The following entities are considered for the adversary model:

- External observers may inspect public blockchain data, IPFS content identifiers, activation records, evidence submissions, and payments. They may also control ordinary user accounts or appliances, but cannot break cryptographic primitives considered secure or extract keys from honest appliance HSMs.
- The VPP is honest-but-curious with respect to privacy: it follows the protocol but may try to infer user behavior from the data it receives.
- The RA is trusted to perform registration and pseudonym rotation correctly and not to disclose the user-pseudonym mapping. The RA and VPP are assumed not to collude.
- Manufacturers and OpenADR certificate authorities are trusted to certify genuine appliances and VEN credentials.

The privacy guarantees of the proposed protocol concern application-layer information. Authenticated and encrypted communication channels can prevent a network adversary from reading or undetectably modifying protected protocol messages, while the freshness and replay checks described in Section 3 prevent reuse of previously valid messages. However, transport-layer protection does not hide metadata such as source IP addresses, communication timing, or traffic volume. A sufficiently powerful network observer could therefore use traffic analysis to weaken the unlinkability guarantees of P2 and P3. In deployments where this threat is relevant, the protocol can be combined with network-layer mechanisms such as VPNs or privacy-preserving relays to reduce source correlation. Resistance to global traffic analysis and denial-of-service attacks based on message dropping or delaying remains outside the scope of this work.

Requirements The proposed system must satisfy a set of requirements. We classify these requirements into three groups: functional requirements (F), privacy requirements (P), and security requirements (S).

F1) *Registration and clustering* Users must be able to register themselves and enroll controllable appliances in the DR program, and each registered appliance must be assigned to a flexibility cluster according to its technical and operational parameters.
F2) *Scheduling* The VPP must be able to compute cluster-level availability to perform the scheduling [12].
P1) *Availability confidentiality.* External observers must not learn the availability schedule of an appliance.
P2) *User-availability unlinkability.* External observers and the VPP must not be able to link a real user identity to a submitted availability.
P3) *Appliance activation privacy.* Activated appliances must not be linkable to a specific user by unauthorized parties.
S1) *Activation authenticity and integrity.* Appliances must accept only authentic and integrity-protected activation events issued by the VPP.
S2) *Replay protection.* Previous activation events and delivery evidence must not be reusable to trigger unauthorized curtailment or duplicate remuneration.
S3) *Valid evidence.* A valid delivery evidence must be from an authorized appliance, bound to a scheduled activation event and unforgeable.
S4) *Payment authorization.* Remuneration must be released only once and only to scheduled appliances that submit valid delivery evidence.

3 Privacy-Preserving Demand Response Scheme

The protocol is divided into a *setup phase*, a *registration phase*, and a *operational phase*.

3.1 Setup Phase

This phase is executed once when the system is set up. It initializes the cryptographic infrastructure, deploys the smart contracts, and defines the components required for the correct execution of the protocol.

VPP Initialization The VPP generates an asymmetric key pair $(sk_{\mathsf{VPP}}, pk_{\mathsf{VPP}})$ and obtains a certificate $\mathsf{Cert}^{\mathsf{CA}}_{\mathsf{VPP}}$ issued by a trusted Certificate Authority CA. This certificate is used to authenticate the VPP in the OpenADR-based demand response signaling infrastructure, where the VPP acts as a Virtual Top Node (VTN) [2].

RA Initialization The RA generates an asymmetric key pair $(sk_{\mathsf{RA}}, pk_{\mathsf{RA}})$, which is later used to encrypt pseudonyms that need to change. It also possesses a certificate $\mathsf{Cert}^{\mathsf{CA}}_{\mathsf{RA}}$ issued by a trusted authority.

Manufacturer Initialization When a new appliance $a_{i,j}$ is manufactured, it is equipped with a Hardware Secure Module (HSM). The HSM generates an unexportable private signing key together with its corresponding public key. Thus, each appliance $a_{i,j}$ is provisioned with an EdDSA signing key pair $(sk^{\mathsf{sig}}_{i,j}, pk^{\mathsf{sig}}_{i,j})$.

The public key $pk^{\mathsf{sig}}_{i,j}$ is certified to authenticate the appliance as a Virtual End Node (VEN) with $\mathsf{Cert}^{\mathsf{CA}}_{\mathsf{VEN},i,j}$ issued by an OpenADR Certificate Authority CA. This certificate is later used by the appliance to authenticate itself as an OpenADR Virtual End Node (VEN) when communicating with the VPP, which acts as the Virtual Top Node (VTN).

3.2 Registration Phase

Each user U_i participating in the demand response program interacts with the system through a mobile application. The mobile application is used to create the user account, configure appliance preferences, and assist in the onboarding of new appliances.

Once the user has been authenticated, the VPP authorizes U_i to register appliances through the Registration Authority (RA) and also sends $Cert^{CA}_{VPP}$. From this point onward, communications between U_i, the RA, and the VPP are performed through authenticated secure channels.

Appliance Registration When a new appliance $a_{i,j}$ is installed, the user executes a secure onboarding procedure through the mobile application.

First, the appliance $a_{i,j}$ enters *pairing mode* after first boot or after a factory reset. Then, U_i, through the mobile application, securely connects with $a_{i,j}$ and performs the following steps:

1. Collects information required to contact the VPP as well as $Cert^{CA}_{VPP}$, and the $\mathsf{Cert}^{\mathsf{CA}}_{\mathsf{RA}}$.
2. Defines the appliance parameter vector $\theta_{i,j} = (max_{i,j},\ rec_{i,j})$.
3. Sends $\theta_{i,j}, Cert^{CA}_{VPP}, \mathsf{Cert}^{\mathsf{CA}}_{\mathsf{RA}}$ to $a_{i,j}$.

After the local pairing procedure, $a_{i,j}$ performs the following operations to obtain the first pseudonym $r = 0$. This pseudonym is renewed in the Operational Phase:

1. Stores the VPP and RA communication parameters.
2. Generates, inside its HSM, a fresh operational signing key pair $\left(sk^{\mathsf{op}}_{i,j,r}, pk^{\mathsf{op}}_{i,j,r}\right)$.
3. Generates a blockchain address $\mathsf{Addr}_{i,j,r}$.
4. Constructs the registration message $m^{\mathsf{reg}}_{i,j,r} = \left[P_{i,j},\ \theta_{i,j},\ pk^{\mathsf{op}}_{i,j,r},\ \mathsf{Addr}_{i,j,r}\right]$.
5. Signs the registration message: $\sigma^{\mathsf{reg}}_{i,j,r} = \mathsf{Sign}_{sk^{\mathsf{sig}}_{i,j}}\left(m^{\mathsf{reg}}_{i,j,r}\right)$.
6. Sends $\left(\mathsf{Cert}^{\mathsf{CA}}_{\mathsf{VEN},i,j},\ m^{\mathsf{reg}}_{i,j,r},\ \sigma^{\mathsf{reg}}_{i,j,r}\right)$ to the RA.

Upon receiving the appliance registration information, the RA performs the following steps:

1. Verifies the validity of the OpenADR VEN certificate $\mathsf{Cert}^{\mathsf{CA}}_{\mathsf{VEN},i,j}$.
2. Verifies the appliance signature: $\mathsf{Verify}_{pk^{\mathsf{sig}}_{i,j}}\left(m^{\mathsf{reg}}_{i,j,r}, \sigma^{\mathsf{reg}}_{i,j,r}\right) = 1$.
3. Checks that $\mathsf{Addr}_{i,j,r}$ and $pk^{\mathsf{op}}_{i,j,r}$ have not been previously enrolled, and marks them as used in its local registry.
4. Assigns $a_{i,j}$ to a cluster according to its technical and operational parameters: $C_c \leftarrow \mathsf{AssignCluster}\left(P_{i,j}, \theta_{i,j}\right)$.
5. Registers $\mathsf{Addr}_{i,j,r}$ and $pk^{\mathsf{op}}_{i,j,r}$ as an authorized operational pseudonym for cluster c in the corresponding $\mathcal{CSC}_c$.
6. Sends the assigned cluster c, the address of $\mathcal{CSC}_c$, and the necessary OpenADR registration parameters to $a_{i,j}$ and U_i.

At the end of this phase, $a_{i,j}$ is cryptographically bound to U_i, assigned to a cluster c, registered as an OpenADR VEN, and authorized to interact with the corresponding $\mathcal{CSC}_c$ under the pseudonym $\left(\mathsf{Addr}_{i,j,r}, pk^{\mathsf{op}}_{i,j,r}\right)$.

Credential lifecycle Operational appliance keys are renewed through the pseudonym-rotation procedure described in Section 3.3. Long-term VPP, RA, and appliance certificates follow the corresponding CA renewal and revocation procedures. If an appliance credential is compromised, its operational pseudonym is deauthorized and the appliance must be securely reprovisioned and re-enrolled.

3.3 Operational Phase

This daily procedure is executed to collect appliance availability and support the scheduling optimization performed by the VPP. The protocol is executed independently for each registered appliance $a_{i,j}$ before the scheduling decision for day d is computed. The number of time slots in one day is defined as T.

Step 1 - Availability Submission The user U_i, through the mobile application, specifies the availability of $a_{i,j}$ for day d:

1. Selects the availability schedule of $a_{i,j}$ for day d.
2. Constructs the availability bitmap $b_{i,j,d} \in \{0,1\}^T$, where 1 indicates that $a_{i,j}$ is available for curtailment during time slot t, and 0 indicates that it is not available.
3. Sends $b_{i,j,d}$ to $a_{i,j}$.

When $a_{i,j}$ receives the information, it performs the following steps:

1. Stores $b_{i,j,d}$ locally and encrypts it under the VPP public key pk_{VPP} using a probabilistic public-key encryption scheme $B_{i,j,d} = \mathsf{Enc}_{pk_{\mathsf{VPP}}}\left(b_{i,j,d}\right)$.
2. Uploads $B_{i,j,d}$ to IPFS and obtains the content identifier $\mathsf{cid}_{i,j,d}$.
3. Using the blockchain address $\mathsf{Addr}_{i,j,r}$ assigned during appliance registration, submits $(d,\ \mathsf{cid}_{i,j,d})$ to $\mathcal{CSC}_c$.

Step 2 - Scheduling For each cluster c and day d, the VPP computes the aggregate availability and determines the activation intervals. The VPP performs the following operations:

1. Retrieves from the $\mathcal{CSC}_c$ all availability submissions registered for day d.
2. Fetches from IPFS all the encrypted payloads referenced by $\mathsf{cid}_{i,j,d}$.
3. Decrypts each encrypted payload using sk_{VPP} and recovers the corresponding set of availability bitmaps $b_{i,j,d}$.
4. Computes the number of available $a_{i,j}$ in each t $N_{c,d}[t] = \sum_{a_{i,j} \in c} b_{i,j,d}[t]$.
5. Computes the aggregate curtailable power available in each time slot $A_{c,d}[t] = N_{c,d}[t] \cdot P_c$.

After computing the aggregate availability, the VPP solves the scheduling optimization problem [12] according to the market request, the available cluster capacity, and the operational constraints of the appliances. The optimization problem is outside the scope of this work.

Let $\mathcal{I}_{c,d}$ denote the set of intervals scheduled for cluster c on day d, which is the output of the scheduling problem. Each scheduled interval is represented as $I = \{t_{\mathsf{ini}}, t_{\mathsf{fin}}\}$. For each scheduled interval $I \in \mathcal{I}_{c,d}$, the VPP performs the following steps:

1. Selects the set of appliance addresses that are available during the whole interval $\alpha_{c,d,I} = \{\mathsf{Addr}_{i,j,r} \ : \ b_{i,j,d}[t] = 1, \ \forall t \in I\}$.
2. Registers $\alpha_{c,d,I}$ in $\mathcal{CSC}_c$: $\mathsf{Schedule}_{c,d,I} = (d, \ I, \ \alpha_{c,d,I})$.

Step 3 - Activation When the scheduled flexibility must be activated, the VPP acts as the OpenADR VTN and issues an OpenADR event directly to the registered appliance VENs. For each activated interval I of cluster c, the VPP performs the following steps:

1. Generates a fresh OpenADR event identifier $\mathsf{eventID}$, associated with the activation event.
2. Defines the activation event metadata: $e_{c,d,I} = (c, \ d, \ I, \ \mathsf{eventID})$.
3. Constructs an OpenADR event message as a VTN:

$$\mathsf{OADR}^{\mathsf{evt}}_{c,d,I} = (\mathsf{eventID}, \ \mathsf{modificationNumber}, \ \mathsf{marketContext}, \\ \mathsf{eiActivePeriod}(I), \ \mathsf{eiEventSignal}(s), \ \mathsf{eiTarget}(c)).$$

4. Sends the OpenADR event directly to the registered appliance VENs using the OpenADR event-distribution service: $\mathsf{oadrDistributeEvent}\left(\mathsf{OADR}^{\mathsf{evt}}_{c,d,I}\right)$.
5. Computes a hash of the OpenADR event: $h^{\mathsf{evt}}_{c,d,I} = H\left(\mathsf{OADR}^{\mathsf{evt}}_{c,d,I}\right)$.
6. Records the activation in the $\mathcal{CSC}_c$ by storing: $(\mathsf{eventID}, \ d, \ I, \ h^{\mathsf{evt}}_{c,d,I})$.

Upon receiving the OpenADR event, each $a_{i,j}$ acts as a VEN and processes the event according to the OpenADR event-handling procedure. In particular, $a_{i,j}$ performs the following steps:

1. Verifies the authenticity of the VTN through $\mathsf{Cert}^{\mathsf{CA}}_{\mathsf{VPP}}$.
2. Checks that the OpenADR $\mathsf{eventID}$ has not already been processed.
3. Extracts $\mathsf{eiTarget}(c)$ and verifies that it matches the locally stored cluster assignment.
4. Checks that its $\mathsf{Addr}_{i,j,r}$ belongs to $\alpha_{c,d,I}$ recorded in $\mathcal{CSC}_c$.
5. Verifies that the activation interval I is compatible with its locally stored availability bitmap $b_{i,j,d}$.
6. If all checks succeed, curtails its power consumption for the full duration of interval I.

Step 4 - Evidence After the activation interval I has finished, each appliance $a_{i,j}$ that has successfully curtailed its consumption for the full duration of the interval generates a delivery attestation. This attestation is used to prove that $a_{i,j}$ participated in the activation event and to enable subsequent verification and reward allocation. If an appliance fails to execute the requested curtailment, it cannot produce valid delivery evidence and is therefore not remunerated. Each appliance $a_{i,j}$ performs the following steps:

1. Verifies that it remained curtailed during the activation interval I.
2. Generates an operational key pair $(sk^{\mathsf{op},r+1}_{i,j}, pk^{\mathsf{op},r+1}_{i,j})$ inside its HSM and derives a blockchain address $\mathsf{Addr}^{r+1}_{i,j}$. The appliance then constructs a rotation request encrypted under the RA public encryption key $C^{\mathsf{rot}}_{i,j,d,I} = \mathsf{Enc}_{pk_{\mathsf{RA}}}(\mathsf{Addr}^{r+1}_{i,j}, pk^{\mathsf{op},r+1}_{i,j})$.
3. Constructs the delivery attestation message
$$m^{\mathsf{del}}_{i,j,d,I} = \left(\mathsf{eventID},\ d,\ I,\ \mathsf{Addr}_{i,j,r},\ C^{\mathsf{rot}}_{i,j,d,I}\right).$$
4. Computes the corresponding attestation digest and signs it
$$h^{\mathsf{del}}_{i,j,d,I} = H\left(m^{\mathsf{del}}_{i,j,d,I}\right), \quad \sigma^{\mathsf{del}}_{i,j,d,I} = \mathsf{Sign}_{sk^{\mathsf{op}}_{i,j,r}}\left(h^{\mathsf{del}}_{i,j,d,I}\right).$$
5. Submits the evidence tuple
$$\pi^{\mathsf{del}}_{i,j,d,I} = \left(m^{\mathsf{del}}_{i,j,d,I},\ h^{\mathsf{del}}_{i,j,d,I},\ \sigma^{\mathsf{del}}_{i,j,d,I}\right)$$
directly to the corresponding cluster smart contract $\mathcal{CSC}_c$ using the appliance blockchain address $\mathsf{Addr}_{i,j,r}$.

Step 5 - Payment After receiving $\pi^{\mathsf{del}}_{i,j,d,I}$, $\mathcal{CSC}_c$ performs the following steps:

1. Checks that an activation record exists for $\mathsf{eventID}$, day d, and interval I.
2. Checks that $\mathsf{Addr}_{i,j,r}$ belongs to the scheduled set $\alpha_{c,d,I}$.
3. Verifies that $m^{\mathsf{del}}_{i,j,d,I}$ refers to the expected activation event, day, interval, and appliance address.
4. Retrieves the operational public key $pk^{\mathsf{op}}_{i,j,r}$ registered for $\mathsf{Addr}_{i,j,r}$.
5. Verifies the appliance signature: $\mathsf{Verify}_{pk^{\mathsf{op}}_{i,j,r}}\left(h^{\mathsf{del}}_{i,j,d,I},\ \sigma^{\mathsf{del}}_{i,j,d,I}\right)$.
6. Checks that $\mathsf{Addr}_{i,j,r}$ has not already been paid for the same activation event.
7. Credits the predefined reward to the appliance address $\mathsf{Addr}_{i,j,r}$.
8. Marks $\mathsf{Addr}_{i,j,r}$ as paid for $\mathsf{eventID}$.

Req.	Main mechanism	Section 3
P1	Encryption of availability before public storage	Step 1
P2	Operational pseudonyms	Registration, Step 1
P3	Operational pseudonyms, pseudonym rotation	Steps 3–5, rotation
S1	VPP authentication, event hash	Step 3
S2	Fresh event identifiers and one-payment state	Steps 3 and 5
S3	Registered operational keys and signed delivery evidence	Registration, Steps 4–5
S4	Schedule membership, evidence verification	Step 5

Table 1. Traceability between requirements and protocol mechanisms.

Pseudonym rotation The RA periodically retrieves pending rotation ciphertexts and performs the following steps:

1. Decrypts them using sk_{RA} and obtains $(\mathsf{Addr}_{i,j}^{r+1}, pk_{i,j}^{\mathsf{op},r+1})$.
2. Verifies that the old address $\mathsf{Addr}_{i,j}^{r}$ is currently authorized in $\mathcal{CSC}_c$, that the new address $\mathsf{Addr}_{i,j}^{r+1}$ has not been previously enrolled, and that the request refers to a valid accepted delivery event.
3. If all checks succeed, the RA authorizes $\mathsf{Addr}_{i,j}^{r+1}$ and $pk_{i,j}^{\mathsf{op},r+1}$ in $\mathcal{CSC}_c$ for the same cluster c.
4. The old address $\mathsf{Addr}_{i,j}^{r}$ is then deactivated for future scheduling epochs.

The RA performs authorizations and deactivations in batches and does not publish an explicit old and new mapping on chain.

4 Discussion

This section evaluates the proposed architecture with respect to the functional, privacy, and security requirements defined in Section 2.3.

Table 1 summarizes where the main mechanisms supporting each privacy and security requirement are introduced in the protocol.

4.1 Functional Feasibility

The proposed architecture provides the functional capabilities required for appliance DR by relying on the protocol phases defined in Section 3. Requirement F1 is addressed by the setup and registration procedures described in Sections 3.1 and 3.2, where the necessary cryptographic material, appliance credentials, operational pseudonyms, and cluster memberships are established before any DR operation is executed. These procedures ensure that only valid appliances are enrolled and that each appliance is associated with the appropriate cluster smart contract.

Requirement F2 is addressed by the scheduling phase described in Section 3.3. In particular, the availability-submission and scheduling steps enable the VPP to derive cluster-level flexibility from appliance availability reports, while the activation step allows the VPP to issue DR events to the selected appliances. The subsequent evidence and payment step provides the mechanism through

which completed activations are verified and remunerated. Therefore, the protocol phases jointly provide the registration, clustering, scheduling, activation, verification, and payment functionality required to support the privacy and security properties analyzed below.

4.2 Security and Privacy Discussion

P1 - Availability confidentiality. P1 requires that external observers do not learn the availability submitted by an appliance. This is achieved by separating the public availability record from its content. The blockchain stores only the submission day, the operational pseudonym, and the IPFS content identifier, whereas the availability bitmap is encrypted under pk_{VPP} using probabilistic public-key encryption before being uploaded to IPFS. Therefore, public blockchain observers, other users, other appliances, and unauthorized third parties may observe that a pseudonymous appliance submitted availability, but cannot recover the submitted time slots. Thus, P1 is satisfied under the confidentiality of the encryption scheme.

P2 - User-availability unlinkability. P2 requires that external observers and the VPP cannot link a real user identity to a submitted availability report. For external observers, this follows from the use of operational pseudonyms: availability submissions are recorded in CSC_c under appliance addresses rather than real identities, and the user-pseudonym mapping is not published on chain. The VPP has stronger access because it decrypts availability bitmaps to compute the schedule. Hence, the protocol does not hide appliance-level availability from the VPP; instead, it limits the VPP to pseudonym-level availability knowledge. Since the user-pseudonym mapping is held by the RA and the VPP and RA are assumed not to collude, the VPP cannot directly associate a submitted availability bitmap with a real user. Therefore, P2 is satisfied within the stated trust model.

P3 - Appliance activation privacy. P3 requires that activated appliances are not linkable to specific users by unauthorized parties. Activation records, scheduled sets, evidence submissions, and payments are expressed through operational pseudonyms rather than real user identities. Consequently, external observers may learn that a given pseudonym participated in an activation, but cannot directly infer the user behind it. The VPP necessarily knows the activated cluster and the scheduled pseudonyms, but does not possess the RA-held user-pseudonym mapping. Pseudonym rotation further reduces long-term linkability: the new address and operational public key are encrypted under pk_{RA} and processed by the RA without publishing an explicit old-new mapping. Rotations are processed in batches to reduce timing and cardinality leakage. Thus, P3 is satisfied against external observers and an honest-but-curious VPP under the non-collusion assumption.

S1 - Activation authenticity and integrity. S1 requires appliances to accept only authentic and integrity-protected activation events issued by the VPP. The VPP acts as the OpenADR VTN and authenticates with $Cert_{VPP}^{CA}$. Upon receiving an event, the appliance verifies the VTN certificate, checks that the $eventID$ has not already been processed, verifies that the target cluster matches its local cluster assignment, and checks that its operational address belongs to the scheduled set $\alpha_{c,d,I}$ stored in CSC_c. Integrity is additionally supported by the on-chain activation hash: the appliance verifies that $H(OADR_{c,d,I}^{evt}) = h_{c,d,I}^{evt}$. These checks bind the activation to the certified VPP, the scheduled interval, the target cluster, and the on-chain schedule, preventing forged or modified activation events unless VPP credentials or cryptographic primitives are compromised.

S2 - Replay protection. S2 requires that previous activation events and delivery evidence cannot be reused to trigger unauthorized curtailment or duplicate remuneration. Activation replay is mitigated because each OpenADR event contains an $eventID$, d and I, and appliances reject identifiers that have already been processed. Evidence replay is prevented by CSC_c, since each delivery attestation is bound to a specific $eventID$, day d, interval I, and operational address $Addr_{i,j,r}$. The contract also checks that the submitting address has not already been paid for the same event and marks it as paid after successful verification. Therefore, replayed activation messages and reused delivery attestations are rejected.

S3 - Valid evidence. S3 requires delivery evidence to be signed with an authorized appliance key and bound to a scheduled activation event. During registration, the RA registers $pk_{i,j,r}^{op}$ and $Addr_{i,j,r}$ in the corresponding CSC_c. After an activation interval, the appliance signs a delivery attestation containing the $eventID$, day, interval, operational address, and optional encrypted rotation request. The contract accepts the evidence only if the claimed activation exists, the submitting address belongs to $\alpha_{c,d,I}$, the attestation refers to the expected event, $h_{i,j,d,I}^{del} = H(m_{i,j,d,I}^{del})$, and the signature verifies under the registered operational public key. These checks reject evidence from unauthorized keys, unscheduled pseudonyms, or different activation events. The link between signed evidence and physical curtailment relies on the trusted appliance control and measurement component connected to the HSM.

S4 - Payment authorization. S4 requires remuneration to be released only once and only to scheduled appliances that submit valid delivery evidence. This is enforced by CSC_c: before authorizing payment, the contract verifies that the activation record exists, the submitting address belongs to the scheduled set, the evidence refers to the expected event, the hash and signature are valid, and the address has not already been paid for the same activation. If any check fails, the evidence is rejected and no remuneration is released. Otherwise, the reward is credited to the operational address and that address is marked as paid for the event.

5 Conclusions

This paper presented a privacy-preserving and secure architecture for appliance-level demand response in Virtual Power Plants. The proposed scheme supports appliance registration, clustering, availability reporting, activation, evidence verification, and payment through the combined use of OpenADR, operational pseudonyms, cryptographic signatures, encrypted availability reports, and cluster smart contracts.

The architecture enables the VPP to compute aggregate cluster-level flexibility and activate scheduled appliances, while limiting the disclosure of user-appliance relationships to unauthorized actors. Moreover, signed activation evidence allows the smart contract to verify participation before remuneration, reducing the risk of forged compensation claims. The discussion shows that the proposed protocol satisfies the functional, privacy, and security requirements defined for the system.

Future work will focus on implementing the architecture, evaluating its computational and communication costs, and integrating the protocol with concrete VPP scheduling algorithms.

Acknowledgments. This research is supported by Project PID2024-156914OB-C42 "SAFE/DATA", funded by MICIU/AEI/ 10.13039/501100011033 and FEDER, UE. Joan Ferré-Queralt is supported by the Spanish Government under an FPU grant (ref. FPU23/01275). Alexandre Viejo is a Serra-Húnter Fellow.

References

1. Adhikari, R., Pipattanasomporn, M., Rahman, S.: An algorithm for optimal management of aggregated hvac power demand using smart thermostats. Applied Energy **217**, 166–177 (2018)
2. Alliance, O.: Openadr 2.0 b profile specification. Raportti. OpenADR Alliance (2013)
3. Di Silvestre, M.L., Gallo, P., Sanseverino, E.R., Sciume, G., Zizzo, G.: Aggregation and remuneration in demand response with a blockchain-based framework. IEEE Transactions on Industry Applications **56**(4), 4248–4257 (2020)
4. Erdinc, O., Taşcıkaraoğlu, A., Paterakis, N.G., Eren, Y., Catalão, J.P.: End-user comfort oriented day-ahead planning for responsive residential hvac demand aggregation considering weather forecasts. IEEE Transactions on Smart Grid **8**(1), 362–372 (2016)
5. Gehbauer, C., Lee, E.S., Wang, T.: An evaluation of the demand response potential of integrated dynamic window and hvac systems. Energy and Buildings **298**, 113481 (2023)
6. Kirby, B.J.: Load response fundamentally matches power system reliability requirements. In: 2007 IEEE power engineering society general meeting. pp. 1–6. IEEE (2007)
7. Kundur, P., et al.: Power system stability. Power system stability and control **10**(1), 7–1 (2007)

8. Li, H., Lin, X., Yang, H., Liang, X., Lu, R., Shen, X.: Eppdr: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid. IEEE Transactions on Parallel and Distributed Systems **25**(8), 2053–2064 (2013)
9. Ma, R., Yi, Z., Xiang, Y., Shi, D., Xu, C., Wu, H.: A blockchain-enabled demand management and control framework driven by deep reinforcement learning. IEEE Transactions on Industrial Electronics **70**(1), 430–440 (2022)
10. Ming, H., Xie, L., Campi, M.C., Garatti, S., Kumar, P.: Scenario-based economic dispatch with uncertain demand response. IEEE Transactions on Smart Grid **10**(2), 1858–1868 (2017)
11. Njoka, G.M., Mogaka, L., Wangai, A.: Impact of variable renewable energy sources on the power system frequency stability and system inertia. Energy Reports **12**, 4983–4997 (2024)
12. Orduz, N., Tsuji, T., Ferré-Queralt, J., Castellà-Roca, J., Viejo, A.: A technical score and cryptography enabled framework for multi market virtual power plant demand response. In: 32nd International Council on Electrical Engineering Conference (ICEE 2026). Seoul, Korea (2026)
13. Parvania, M., Fotuhi-Firuzabad, M., Shahidehpour, M.: Optimal demand response aggregation in wholesale electricity markets. IEEE transactions on smart grid **4**(4), 1957–1965 (2013)
14. Parvania, M., Fotuhi-Firuzabad, M., Shahidehpour, M.: Iso's optimal strategies for scheduling the hourly demand response in day-ahead markets. IEEE Transactions on Power Systems **29**(6), 2636–2645 (2014)
15. Rottondi, C., Verticale, G.: Privacy-friendly load scheduling of deferrable and interruptible domestic appliances in smart grids. Computer Communications **58**, 29–39 (2015)
16. Singh, S., Kumar, A.: Demand response program solution to manage congestion in transmission network considering uncertainty of load. In: 2017 8th international conference on computing, communication and networking technologies (ICCCNT). pp. 1–6. IEEE (2017)
17. Venizelou, V., Tsolakis, A.C., Evagorou, D., Patsonakis, C., Koskinas, I., Theraapontos, P., Zyglakis, L., Ioannidis, D., Makrides, G., Tzovaras, D., et al.: Dso-aggregator demand response cooperation framework towards reliable, fair and secure flexibility dispatch. Energies **16**(6), 2815 (2023)
18. Wei, M., Zhong, J.: Optimal bidding strategy for demand response aggregator in day-ahead markets via stochastic programming and robust optimization. In: 2015 12th International Conference on the European Energy Market (EEM). pp. 1–5. IEEE (2015)
19. Yang, Q., Wang, H., Wang, T., Zhang, S., Wu, X., Wang, H.: Blockchain-based decentralized energy management platform for residential distributed energy resources in a virtual power plant. Applied Energy **294**, 117026 (2021)
20. Yi, Z., Xu, Y., Gu, W., Wu, W.: A multi-time-scale economic scheduling strategy for virtual power plant based on deferrable loads aggregation and disaggregation. IEEE Transactions on Sustainable Energy **11**(3), 1332–1346 (2019)
21. Yolcan, O.O.: World energy outlook and state of renewable energy: 10-year evaluation. Innovation and Green Development **2**(4), 100070 (2023)
22. Yu, H., Zhang, J., Ma, J., Chen, C., Geng, G., Jiang, Q.: Privacy-preserving demand response of aggregated residential load. Applied Energy **339**, 121018 (2023)
23. Zhang, X., Hug, G.: Bidding strategy in energy and spinning reserve markets for aluminum smelters' demand response. In: 2015 IEEE power & energy society innovative smart grid technologies conference (ISGT). pp. 1–5. IEEE (2015)